Oat v. 1 Language Specification

CIS3410 – Steve Zdancewic

March 2, 2024

1 Grammar

The following grammar defines the Oat syntax. All binary operations are *left associative* with precedence levels indicated numerically. Higher precedence operators bind tighter than lower precedence ones.

| *prog* | ::= | | prog |
| | | $decl_1 .. decl_i$ | |

decl	::=		global declarations
		gdecl	
		fdecl	

| *gdecl* | ::= | | global variable declarations |
| | | global *id* = *gexp*; | |

| *arg* | ::= | | arg |
| | | *t id* | |

| *args* | ::= | | args |
| | | $arg_1, .. , arg_n$ | |

| *fdecl* | ::= | | function declaration |
| | | *retty id*(*args*) *block* | |

| *block* | ::= | | blocks |
| | | $\{stmt_1 .. stmt_n\}$ | |

t	::=		types
		int	
		bool	
		ref	

ref	::=		reference types
		string	
		t[]	

F	::=		function types
	|	$(t_0, .., t_n) \rightarrow retty$	

retty	::=		return types
	|	`void`	
	|	*t*	

bop	::=		(left associative) binary operations
	|	`*`	multiplication (precedence 100)
	|	`+`	addition (precedence 90)
	|	`-`	subtraction (precedence 90)
	|	`<<`	shift left (precedence 80)
	|	`>>`	shift right logical (precedence 80)
	|	`>>>`	shift right arithmetic (precedence 80)
	|	`<`	less-than (precedence 70)
	|	`<=`	less-than or equal (precedence 70)
	|	`>`	greater-than (precedence 70)
	|	`>=`	greater-than or equal (precedence 70)
	|	`==`	equal (precedence 60)
	|	`!=`	not equal (precedence 60)
	|	`&`	logical and (precedence 50)
	|	`|`	logical or (precedence 40)
	|	`[&]`	bit-wise and (precedence 30)
	|	`[|]`	bit-wise or (precedence 20)

uop	::=		unary operations
	|	`-`	
	|	`!`	
	|	`~`	

gexp	::=		global initializers
	|	*integer*	64-bit integer literals
	|	*string*	C-style strings
	|	*ref* `null`	
	|	`true`	
	|	`false`	
	|	`new` t`[]{`$gexp_1, .., gexp_n$`}`	

lhs	::=		lhs expressions
	|	*id*	
	|	exp_1`[`exp_2`]`	

exp	::=		expressions
	|	*id*	
	|	*integer*	64-bit integer literals
	|	*string*	C-style strings
	|	*ref* null	
	|	true	
	|	false	
	|	$exp_1 [exp_2]$	
	|	$id(exp_1, .., exp_n)$	
	|	new t[]{$exp_1, .., exp_n$}	Explicitly initialized array
	|	new int [exp_1]	Default-initialize int array
	|	new bool [exp_1]	Default-initialize bool array
	|	exp_1 *bop* exp_2	
	|	*uop* *exp*	
	|	(*exp*)	

vdecl	::=		local declarations
	|	var *id* = *exp*	

vdecls	::=		decl list
	|	$vdecl_1, .., vdecl_n$	

stmt	::=		statements
	|	*lhs* = *exp*;	
	|	*vdecl*;	
	|	return *exp*;	
	|	return ;	
	|	$id(exp_1, .., exp_n)$;	
	|	*if_stmt*	
	|	for(*vdecls*; exp_{opt}; $stmt_{opt}$) *block*	
	|	while(*exp*) *block*	

if_stmt	::=		if statements
	|	if(*exp*) *block* *else_stmt*	

else_stmt	::=		else
	|	ϵ	
	|	else *block*	
	|	else *if_stmt*	

2 Typing Rules

$\boxed{\vdash bop_1, .., bop_i : F}$

$$\frac{}{\vdash \texttt{+,*,-,<<,>>,>>>,[\&],[|]} : (\texttt{int},\texttt{int}) \rightarrow \texttt{int}} \quad \text{TYP\_INTOPS}$$

$$\frac{}{\vdash \texttt{==,!=,<,<=,>,>=} : (\texttt{int},\texttt{int}) \rightarrow \texttt{bool}} \quad \text{TYP\_CMPOPS}$$

$$\frac{}{\vdash \texttt{\&,|} : (\texttt{bool},\texttt{bool}) \rightarrow \texttt{bool}} \quad \text{TYP\_BOOLOPS}$$

$\boxed{\vdash uop : F}$

$$\frac{}{\vdash \texttt{!} : (\texttt{bool}) \rightarrow \texttt{bool}} \quad \text{TYP\_LOGNOT}$$

$$\frac{}{\vdash \texttt{\textasciitilde} : (\texttt{int}) \rightarrow \texttt{int}} \quad \text{TYP\_BITNEG}$$

$$\frac{}{\vdash \texttt{-} : (\texttt{int}) \rightarrow \texttt{int}} \quad \text{TYP\_NEG}$$

$\boxed{G;L \vdash exp : t}$

$$\frac{x{:}t \in L}{G;L \vdash x : t} \quad \text{TYP\_LOCAL}$$

$$\frac{}{G;L \vdash n : \texttt{int}} \quad \text{TYP\_INT}$$

$$\frac{}{G;L \vdash s : \texttt{string}} \quad \text{TYP\_STRING}$$

$$\frac{}{G;L \vdash \texttt{string null} : \texttt{string}} \quad \text{TYP\_NULLSTR}$$

$$\frac{}{G;L \vdash t\texttt{[] null} : t\texttt{[]}} \quad \text{TYP\_NULLARR}$$

$$\frac{}{G;L \vdash \texttt{true} : \texttt{bool}} \quad \text{TYP\_TRUE}$$

$$\frac{}{G;L \vdash \texttt{false} : \texttt{bool}} \quad \text{TYP\_FALSE}$$

$$\frac{G;L \vdash exp_1 : t\texttt{[]} \quad G;L \vdash exp_2 : \texttt{int}}{G;L \vdash exp_1[exp_2] : t} \quad \text{TYP\_INDEX}$$

$$\frac{f{:}(t_1, .., t_i) \rightarrow t \in G \quad G;L \vdash exp_1 : t_1 \quad .. \quad G;L \vdash exp_i : t_i}{G;L \vdash f(exp_1, .., exp_i) : t} \quad \text{TYP\_CALL}$$

$$\frac{G;L \vdash exp_1 : t \quad .. \quad G;L \vdash exp_i : t}{G;L \vdash \texttt{new } t\texttt{[]}\{exp_1, .., exp_i\} : t\texttt{[]}} \quad \text{TYP\_ARRLIT}$$

$$\frac{G;L \vdash exp_1 : \texttt{int}}{G;L \vdash \texttt{new int }[exp_1] : t\texttt{[]}} \quad \text{TYP\_ARRZEROINT}$$

$$\frac{G;L \vdash exp_1 : \texttt{int}}{G;L \vdash \texttt{new bool }[exp_1] : t\texttt{[]}} \quad \text{TYP\_ARRZEROBOOL}$$

$$\frac{\vdash bop : (t_1, t_2) \rightarrow t \quad G;L \vdash exp_1 : t_1 \quad G;L \vdash exp_2 : t_2}{G;L \vdash exp_1 \, bop \, exp_2 : t} \quad \text{TYP\_BOP}$$

$$\frac{\vdash uop : (t) \rightarrow t \quad G;L \vdash exp : t}{G;L \vdash uop \, exp : t} \quad \text{TYP\_UOP}$$

$$\boxed{G\,;L_1 \vdash \mathit{vdecl} \Rightarrow L_2}$$

$$\frac{G\,;L \vdash \mathit{exp} : t \quad x \notin L}{G\,;L \vdash \mathtt{var}\ x = \mathit{exp} \Rightarrow L, x\!:\!t} \quad \text{TYP\_DECL}$$

$$\boxed{G\,;L_0 \vdash \mathit{vdecls} \Rightarrow L_i}$$

$$\frac{G\,;L_0 \vdash \mathit{vdecl}_1 \Rightarrow L_1 \quad \ldots \quad G\,;L_{i-1} \vdash \mathit{vdecl}_i \Rightarrow L_i}{G\,;L_0 \vdash \mathit{vdecl}_1, .., \mathit{vdecl}_i \Rightarrow L_i} \quad \text{TYP\_VDECLS}$$

$$\boxed{G\,;L_1\,;\mathit{retty} \vdash \mathit{stmt} \Rightarrow L_2}$$

$$\frac{G\,;L_1 \vdash \mathit{vdecl} \Rightarrow L_2}{G\,;L_1\,;t \vdash \mathit{vdecl}\,; \Rightarrow L_2} \quad \text{TYP\_SDECL}$$

$$\frac{G\,;L \vdash \mathit{lhs} : t \quad G\,;L \vdash \mathit{exp}_2 : t}{G\,;L\,;t \vdash \mathit{lhs} = \mathit{exp}_2\,; \Rightarrow L} \quad \text{TYP\_ASSN}$$

$$\frac{f\!:\!(t_1, .., t_i) \rightarrow \mathtt{void} \in G \quad G\,;L \vdash \mathit{exp}_1 : t_1 \quad .. \quad G\,;L \vdash \mathit{exp}_i : t_i}{G\,;L\,;t \vdash f(\mathit{exp}_1, .., \mathit{exp}_i)\,; \Rightarrow L} \quad \text{TYP\_SCALL}$$

$$\frac{G\,;L \vdash \mathit{exp} : \mathtt{bool} \quad G\,;L\,;t \vdash \mathit{block}_1 \quad G\,;L\,;t \vdash \mathit{block}_2}{G\,;L\,;t \vdash \mathtt{if}(\mathit{exp})\ \mathit{block}_1\ \mathtt{else}\ \mathit{block}_2 \Rightarrow L} \quad \text{TYP\_IF}$$

$$\frac{G\,;L \vdash \mathit{exp} : \mathtt{bool} \quad G\,;L\,;t \vdash \mathit{block}}{G\,;L\,;t \vdash \mathtt{while}(\mathit{exp})\ \mathit{block} \Rightarrow L} \quad \text{TYP\_WHILE}$$

$$\frac{G\,;L_1 \vdash \mathit{vdecls} \Rightarrow L_2 \quad G\,;L_2 \vdash \mathit{exp} : \mathtt{bool} \quad G\,;L_2\,;t \vdash \mathit{stmt} \Rightarrow L_3 \quad G\,;L_2\,;t \vdash \mathit{block}}{G\,;L_1\,;t \vdash \mathtt{for}(\mathit{vdecls}\,; \mathit{exp}_{opt}\,; \mathit{stmt}_{opt})\ \mathit{block} \Rightarrow L_1} \quad \text{TYP\_FOR}$$

$$\frac{G\,;L \vdash \mathit{exp} : t}{G\,;L\,;t \vdash \mathtt{return}\ \mathit{exp}\,; \Rightarrow L} \quad \text{TYP\_RET}\textsc{t}$$

$$\frac{}{G\,;L\,;\mathtt{void} \vdash \mathtt{return}\,; \Rightarrow L} \quad \text{TYP\_RET}\textsc{Void}$$

$$\boxed{G\,;L\,;t \vdash \mathit{block}}$$

$$\frac{G\,;L_0\,;t \vdash \mathit{stmt}_1 .. \mathit{stmt}_i \Rightarrow L_i}{G\,;L_0\,;t \vdash \{\mathit{stmt}_1 .. \mathit{stmt}_i\}} \quad \text{TYP\_BLOCK}$$

$$\boxed{G\,;L_0\,;t \vdash \mathit{stmt}_1 .. \mathit{stmt}_i \Rightarrow L_i}$$

$$\frac{G\,;L_0\,;t \vdash \mathit{stmt}_1 \Rightarrow L_1 \quad \ldots \quad G\,;L_{i-1}\,;t \vdash \mathit{stmt}_i \Rightarrow L_i}{G\,;L_0\,;t \vdash \mathit{stmt}_1 .. \mathit{stmt}_i \Rightarrow L_i} \quad \text{TYP\_STMTS}$$

$$\boxed{G_0 \vdash decl \Rightarrow G_1}$$

$$\frac{x \notin G \quad \cdot\,;\cdot \vdash gexp : t}{G \vdash \texttt{global } x = gexp\,; \;\Rightarrow G, x{:}t} \quad \textbf{TYP\_VDECL}$$

$$\frac{f \notin G}{G \vdash t\,f(t_1\,x_1, .., t_i\,x_i)\;block \Rightarrow G, f{:}(t_1, .., t_i) \rightarrow t} \quad \textbf{TYP\_FDECL}$$

$$\boxed{G_0 \vdash decl_1 \,..\, decl_i \Rightarrow G_i}$$

$$\frac{G_0 \vdash decl_1 \Rightarrow G_1 \quad \ldots \quad G_{i-1} \vdash decl_i \Rightarrow G_i}{G_0 \vdash decl_1 \,..\, decl_i \Rightarrow G_i} \quad \textbf{TYP\_GLOBAL\_CTXT}$$

$$\boxed{G \vdash decl}$$

$$\frac{G\,;x_1{:}t_1, .., x_i{:}t_i\,;t \vdash block}{G \vdash t\,f(t_1\,x_1, .., t_i\,x_i)\;block} \quad \textbf{TYP\_GFUN}$$

$$\frac{\cdot\,;\cdot \vdash gexp : t}{G \vdash \texttt{global } x = gexp\,;} \quad \textbf{TYP\_GVAR}$$

$$\boxed{\vdash prog}$$

$$\frac{G_0 \vdash decl_1 \,..\, decl_i \Rightarrow G \quad G \vdash decl_1 \quad .. \quad G \vdash decl_i}{\vdash decl_1 \,..\, decl_i} \quad \textbf{TYP\_PROG}$$